Preprint
Accepted for publication 2007

Creating Protein Models from Electron-Density Maps using Particle-Filtering Methods

Frank DiMaio[a,b*], Dmitry A. Kondrashov[c], Eduard Bitto[d], Ameet Soni[a,b], Craig A. Bingman[d], George N. Phillips, Jr.[c,a,d], Jude W. Shavlik[a,b]

[a]Computer Sciences Dept., [b]Biostatistics and Medical Informatics Dept., [c]Biochemistry Dept., [d]Center for Eukaryotic Structural Genomics
University of Wisconsin, Madison, WI, 53706

ABSTRACT

Motivation: One bottleneck in high-throughput protein crystallography is interpreting an electron-density map; that is, fitting a molecular model to the 3D picture crystallography produces. Previously, we developed ACMI, an algorithm that uses a probabilistic model to infer an accurate protein backbone layout. Here we use a sampling method known as particle filtering to produce a set of all-atom protein models. We use the output of ACMI to guide the particle filter's sampling, producing an accurate, physically feasible set of structures.

Results: We test our algorithm on ten poor-quality experimental density maps. We show that particle filtering produces accurate all-atom models, resulting in fewer chains, lower sidechain RMS error, and reduced R factor, compared to simply placing the best-matching sidechains on ACMI's trace. We show that our approach produces a more accurate model than three leading methods – TEXTAL, RESOLVE, and ARP/wARP – in terms of main chain completeness, sidechain identification, and crystallographic R factor.

Availability: Source code and experimental density maps available at *ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/programs/acmi/*.

Contact: dimaio@cs.wisc.edu

1 INTRODUCTION

Knowledge of the spatial arrangement of constituent atoms in a complex biomolecules, such as proteins, is vital for understanding their function. X-ray crystallography is the primary technique for determination of atomic positions, or the *structure*, of biomolecules. A beam of X-rays is diffracted by a crystal, resulting in a set of reflections that contain information about the molecular structure. This information can be interpreted to produce a 3D image of the macromolecule, which is usually represented by an electron-density map. Interpretation of these maps requires locating the atoms in complex three-dimensional images. This is a difficult, time-consuming process, that may require weeks or months of an expert crystallographer's time.

Our previous work (DiMaio *et al.*, 2006) developed the automatic interpretation tool ACMI (Automatic Crystallographic Map Interpreter). ACMI employs probabilistic inference to compute a probability distribution of the coordinates of each amino acid, given the electron-density map. However, ACMI makes several simplifications, such as reducing each amino acid to a single atom and confining the locations to a coarse grid. In this work we introduce the use of a statistical sampling method called *particle filtering* (PF) (Doucet *et al.*, 2000) to construct all-atom protein models, by stepwise extention of a set of incomplete models drawn from a distribution computed by ACMI. This results in a set of probability-weighted all-atom protein models. The method interprets the density map by generating a number of distinct protein conformations consistent with the data. We compare the single model that best matches the density map (without knowing the true solution) with the output of existing automated methods, on multiple sets of crystallographic data which required considerable human effort to solve. We also show that modeling the data with a set of structures, obtained from several particle-filtering runs, results in a better fit than using one structure from a single particle-filtering run. Particle filtering enables the automated building of detailed atomic models for challenging protein crystal data, with a more realistic representation of conformational variation in the crystal.

[*]to whom correspondence should be addressed

2 PROBLEM OVERVIEW AND RELATED WORK

In recent years, considerable investment into structural genomics (i.e. high-throughput determination of protein structures) has yielded a wealth of new data (Berman & Westbrook, 2004; Chandonia & Brenner, 2006). The demand for rapid structure solution is growing, and automated methods are being deployed at all stages of the structural determination process. These new technologies include cell-free methods for protein production (Sawasaki *et al.*, 2002), the use of robotics to screen thousands of crystallization conditions (Snell *et al.*, 2004), and new software for automated building of macromolecular models based on the electron-density map (DiMaio *et al.*, 2006; Morris *et al.*, 2003; Ioerger & Sacchettini, 2003; Terwilliger, 2002; Cowtan, 2006). The last problem is addressed in this study.

2.1 Density-map interpretation

A beam of X-rays scattered by a crystalline lattice produces a pattern of reflections, which are measured by a detector. Given complete information, i.e., both the amplitudes and the phases of the reflected photons, one can reconstruct the electron-density map as the Fourier transform of these complex-valued reflections. However, the detector can only measure the intensities of the reflections and not the phases. Thus, a fundamental problem of crystallography lies in approximating the unknown phases. Our aim is the construction of an all-atom protein model that best fits a given electron-density map based on approximate phasing.

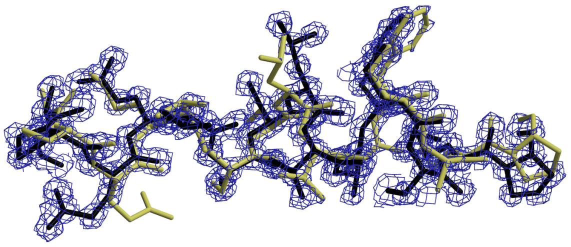

Fig. 1. An overview of density-map interpretation. The density map is illustrated with contours enclosing regions of higher density; the protein model uses sticks to indicate bonds between atoms. This figure shows two protein models fit to the density map, one darker and one lighter.

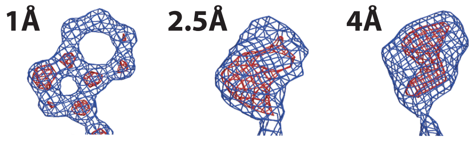

Fig. 2. The effect of varying resolution on electron density of a tryptophan sidechain, with phases computed from a final atomic model. The effects of phase error are similar to worsening the resolution.

The electron-density map is defined on a three-dimensional grid of points covering the *unit cell*, which is the basic repeating unit in the protein crystal. A crystallographer, given the the amino-acid sequence of the protein, attempts to place the amino acids in the unit cell, based on the shape of the electron-density contours. Figure 1 shows the electron-density map as an isocontoured surface. This figure also shows two models of atomic positions consistent with the electron density, where sticks indicate bonds between atoms.

The quality of an electron-density map is limited by its *resolution*, which, at its high limit, corresponds to the smallest interplanar distance between diffracting planes. The highest resolution for a data set depends on the order in the crystalline packing, the detector sensitivity, and the brightness of the X-ray source. Figure 2 illustrates the electron density around a tryptophan sidechain at varying resolution, with "ideal" phases computed from a complete all-atom model. Note that at 1 Å resolution, the spheres of individual atoms are clearly visible, while at 4 Å even the overall shape of the tryptophan sidechain is distorted. Typical resolution for protein structures lies in the 1.5 – 2.5 Å range.

Another factor that affects the quality of an electron-density map is the accuracy of the computed phases. To obtain an initial approximation of the phases, crystallographers use techniques based on the special features in X-ray scattering produced by heavy atoms, such as multiple-wavelength or single-wavelength anomalous diffraction (MAD or SAD) and multiple isomorphous replacement (MIR). This allows the computation of an initial electron-density map, the quality of which greatly depends on the fidelity of the initial phasing. Artifacts produced by phase error are similar to those of worsening resolution; additionally, high spatial frequency noise is also present. The interpretation of a poorly phased map can be very difficult even for a trained expert.

2.2 ACMI's probabilistic protein backbone tracing

We previously developed a method, ACMI, that produces high-confidence backbone traces from poor-quality maps. Given a density map and the protein's amino-acid sequence, ACMI constructs a probabilistic model of the location of each Cα. Statistical inference on this model gives the most probable backbone trace *of the given sequence* in the density map.

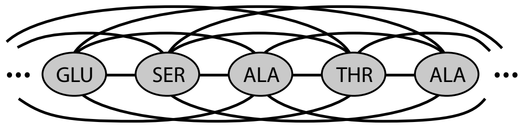

Fig. 3. A sample undirected graphical model corresponding to some protein. The probability of some backbone model is proportional to the product of potential functions: one associated with each vertex, and one with each edge in the fully connected graph.

ACMI models a protein using an *pairwise Markov field*. As illustrated in Figure 3, this approach defines the probability distribution of a set of variables on an undirected graph. Each vertex in the graph is associated with one or more variables, and the probability of some setting of these variables is the product of *potential functions* associated with vertices and edges in the graph.

In ACMI's protein model, vertices correspond to individual amino-acid residues, and the variables associated with each vertex correspond to an amino acid's Cα location and orientation. The vertex potential ψ_i at each node i can be thought of as a "prior probability" on each alpha carbon's location given the density map and ignoring the locations of other amino acids. In this model, the probability of some backbone conformation $\mathbf{B} = \{b_1, \ldots, b_N\}$, given density map $\mathbf{M}$ is given as

$$P(\mathbf{B}|\mathbf{M}) = \prod_{\text{amino-acid } i} \psi_i(b_i|\mathbf{M}) \times \prod_{\text{amino-acids } i,j} \psi_{ij}(b_i, b_j) \quad (1)$$

ACMI's ψ_i considers a 5-mer (5-amino-acid sequence) centered at each position in the protein sequence, and searches a non-redundant subset of the *Protein Data Bank* (PDB) (Wang & Dunbrack, 2003) for observed conformations of that 5-mer, using the computed density (conditioned on the map resolution) of each conformation as a search target. An improvement to our original approach (DiMaio *et al.*, 2007) uses spherical harmonic decomposition to rapidly search over all rotations of each search target.

The edge potentials ψ_{ij} associated with each edge model global spatial constraints on the protein. ACMI defines two types of edge potentials: adjacency constraints ψ_{adj} model interactions between adjacent residues (in the primary sequence), while occupancy constraints ψ_{occ} model interactions between residues distant on the protein chain (though not necessarily spatially distant in the folded structure). Adjacency constraints make sure that Cα's of adjacent amino acids are about 3.8 Å apart; occupancy constraints make sure no two amino acids occupy the same 3D space. Multiple subunits in the asymmetric unit are handled by fully connected each subunit with occupancy-constraining edges.

A fast approximate-inference algorithm finds likely locations of each Cα, given the vertex and edge potentials. For each amino acid in the provided protein sequence, ACMI's inference algorithm returns the marginal distribution of that amino acid's Cα location: that is, the probability distribution taking into account the position of all other amino acids. Our previous work shows that ACMI produces more accurate backbone traces than alternative approaches (DiMaio *et al.*, 2006). Also, ACMI is less prone to missing pieces in the model, because locations of amino acids not visible in the density map are inferred from the locations of neighboring residues.

2.3 Other approaches

Several methods have been developed to automatically interpret electron-density maps. Given high-quality data (up to about 2.3 Å resolution), one widely used algorithm is ARP/wARP (Morris *et al.*, 2003). This atom-based method heuristically places atoms in the map, connects them, and refines their positions. To handle poor phasing, ARP/ wARP alternates steps in which (a) a model is built based on a density map, and (b) the map is improved using phases from iteratively refined model.

Other methods have been developed to handle low-resolution density maps, where atom-based approaches like ARP/wARP fail to produce a reasonable model. Ioerger's TEXTAL (Ioerger & Sacchettini, 2003) and CAPRA (Ioerger & Sacchettini, 2002) interpret poor-resolution density maps using ideas from pattern recognition. Given a sphere of density from an uninterpreted density map, both employ a set of rotation-invariant statistical features to aid in interpretation. CAPRA uses a trained neural network to identify Cα locations. TEXTAL performs a rotational search to place sidechains, using the rotation-invariant features to identify sidechain types. RESOLVE's automated model-building routine (Terwilliger, 2002) uses a hierarchical procedure in which helices and strands are located by an exhaustive search. High-scoring matches are extended iteratively using a library of tripeptides; these growing chains are merged using a heuristic. BUCCANEER (Cowtan, 2006) is a newer probabilistic approach to interpreting poor quality maps; currently, the algorithm only constructs a main chain trace.

At lower resolution and with greater phase error, these methods have difficulty in chain tracing and especially in correctly identifying amino acids. Unlike ACMI's model-based approach, they first build a backbone model, then align the protein sequence to it. At low resolutions, this alignment often fails, resulting in the inability to correctly identify sidechain types. These approaches have a tendency to produce disjointed chains in poor-resolution maps, which requires significant human labor to repair.

3 METHODS

For each amino acid i, ACMI's probabilistic inference returns the marginal probability distribution $\hat{p}_i(b_i)$ of that amino acid's Cα position. Previously, we computed the backbone trace $\mathbf{B} = \{b_1, \ldots, b_N\}$ (where b_i describes the position and rotation of amino-acid i) as the position of each Cα that maximized ACMI's belief,

$$b_i^* = \arg\max_{b_i} \hat{p}_i(b_i) \tag{2}$$

One obvious shortcoming in this previous approach is that biologists are interested in not just the position b_i^* of each Cα, but in the location of every (non-hydrogen) atom in the protein. Naïvely, we could take ACMI's most-probable backbone model, and simply attach the best-matching sidechain from a library of conformations to each of the model's Cα positions. In Section 4 we show that such a method works reasonably well. Another issue is that the marginal distributions are computed on a grid, which may lead to nonphysical distances between residues when Cα's are placed on the nearest grid points. Additionally, ACMI's inference is approximate, and errors due to these approximations may produce an incorrect backbone trace, with two adjacent residues located on opposite sides of the map.

Another problem deals with using a "maximum-marginal backbone trace," that is, independently selecting the position of each residue to maximize the marginal. A density map that contains a mixture of (physically-feasible) protein conformations may have a maximum-marginal conformation that is physically unrealistic. Representing each amino acid's position as a distribution over the map is very expressive. Simply returning the Cα position that maximizes the marginal ignores a lot of information. This section details the application of particle filtering to "explain" the density map using multiple, physically feasible models.

3.1 Particle-filtering overview

We will use a particle-filtering method called statistical importance resampling (SIR) (Doucet *et al.*, 2000; Arulampalam *et al.*, 2001), which approximates the posterior probability distribution of a state sequence $x_{1:K} = \{x_1, \ldots, x_K\}$ given observations $y_{1:K}$ as the weighted sum of a finite number of point estimates $x_{1:K}^{(i)}$,

$$p(x_{1:K}|y_{1:K}) \approx \sum_{i=1}^{N} w_i \delta(x_{1:K} - x_{1:K}^{(i)}) \tag{3}$$

Here, i is the particle index, w_i is particle i's weight, K is the number of states (here the number of amino acids), and δ is the Dirac delta function. In our application, x_k describes the position of every non-hydrogen atom in amino acid k; y_k is a 3D region of density in the map.

In our work, the technical term "particle" refers to one specific 3D layout of all the non-hydrogen atoms in a contiguous subsequence of the protein (e.g., from amino acid 21 to 25). PF represents the distribution of some subsequence's layout using a set of distinct layouts for that subsequence (in other words, what we are doing is illustrated in Figure 1, where each protein model is a single particle).

At each iteration of particle filtering, we advance the extent of each particle by one amino acid. For example, given $x_{21:25} = \{x_{21}, \ldots .x_{25}\}$ the position of all atoms in amino acids 21 through 25 (we will use this shorthand notation for a particle throughout the paper), PF samples the position of the next amino acid, in this case x_{26}. Ideally, particle filtering would sample these positions from the posterior distribution: the probability of x_{26}'s layout given the current particle and the map. SIR is based on the assumption that this posterior is difficult to sample directly, but easy to evaluate (up to proportionality). Given some other function (called the *importance function*) that approximates the posterior, particle filtering samples from this function instead, then uses the ratio of the posterior to the importance function to reweight the particles.

To give an example of an importance function, particle-filtering applications often use the prior *conditional distribution* $p(x_k|x_{k-1})$ as the importance function. After sampling the data, y_k will be used to weight each particle. In our application, this is analogous to placing an amino acid's atoms using only the layout of the previous amino acid, then reweighting by how well it fits the density map.

We use a particle resampling step to address the problem of degeneracy in the particle ensemble (Kong *et al.*, 1994). As particles are extended, the variance of particle weights tends to increase, until there are few particles with non-negligible weights, and much effort is spent updating particles with little or no weight. To ameliorate this problem, an optional resampling step samples (with replacement) a new set of N particles at each iteration, with the probability of selecting a particle proportional to its weight. This ensures most particles remain on high-likelihood trajectories in state space.

What makes SIR (and particle filtering methods in general) different from Markov-chain Monte Carlo (MCMC) is that MCMC is concerned with the stationary distribution of the Markov chain. In particle filtering, one is not concerned with convergence of the point estimates, rather, the distribution is simply modeled by the ensemble of particles, whether or not they converge.

3.2 Protein particle model

An overview of our entire algorithm appears in Algorithm 1. For density-map interpretation, we use the variable x_k to denote the position of every atom in amino-acid k. We want to find the complete (all-atom) protein model $x_{1:K}$ that best explains the observed electron-density map $\mathbf{y}$. To simplify, we parameterize x_k as a Cα location b_k (the same as b_i in Equation 2), and a sidechain placement s_k. The sidechain placement identifies the *3D location of every non-hydrogen sidechain atom* in amino-acid k, as well as the position of backbone atoms C, N, and O.

Algorithm 1: ACMI-PF's algorithm for growing a protein model.

```
input  : density map y, amino-acid marginals $\hat{p}_k(b_k)$
output: set of protein models $x^{(i)}_{1:K}$ and weights $w^{(i)}_K$
// start at some AA with high certainty about its location
choose $k$ such that $\hat{p}_k(b^{(i)}_k)$ has minimum entropy
foreach particle $i = 1 \ldots N$ do
    choose $b^{(i)}_k$ at random from $\hat{p}_k(b^{(i)}_k)$
    $w^{(i)}_k \leftarrow 1/N$
end
foreach residue $k$ do
    foreach particle $i = 1 \ldots N$ do
        // choose $b_{k+1}$ (or $b_{k-1}$) given $b^{(i)}_k$
        $\{b^{*m}_{k+1}\} \leftarrow$ choose $M$ samples from $\phi_{adj}(b^{(i)}_k, b_{k+1})$
        $w^{*m} \leftarrow$ belief $\hat{p}_i(b^{*m}_{k+1})$
        $b^{(i)}_{k+1} \leftarrow$ choose $b^{*m}_{k+1}$ with probability $\propto w^{*m}$
        $w^{(i)}_{k+1} \leftarrow w^{(i)}_k \cdot \sum_{m=1}^{M} w^{*m}$

        // choose $s_k$ given $b^{(i)}_{k-1:k+1}$
        $\{s^{*l}_k\} \leftarrow$ sidechain conformations for amino-acid $k$
        $p^{*l}_{null} \leftarrow$ prob $cc(s^{*l}_k, \text{EDM}[b_k])$ occurred by chance
        $s_k \leftarrow$ choose $s^{*l}_k$ with probability $\propto 1/p^{*l}_{null} - 1$
        $w^{(i)}_{k+1} \leftarrow w^{(i)}_k \cdot \sum_{l=1}^{L} 1/p^{*l}_{null} - 1$
    end
end
```

Given this parameterization, the Markov process alternates between placing: (a) Cα positions and (b) sidechain atoms. That is, an iteration of particle filtering first samples b_{k+1} (Cα of amino-acid $k+1$) given b_k, or alternatively, growing our particle toward the N-terminus would sample b_{k-1} given b_k. Then, given the triple $b_{k-1:k+1}$, we sample sidechain conformation s_k.

3.2.1 Using ACMI*-computed marginals to place Cα's.* In our algorithm's backbone step we want to sample the Cα position b_{k+1} (or b_{j-1}), given our growing trace $b^{(i)}_{j:k}$, for each particle i. That is, we want to define our sampling function $q(b_{k+1}|b^{(i)}_j, \ldots, b^{(i)}_k, \mathbf{y})$. Doucet *et al.* (2000) defines the *optimal sampling function* as the conditional marginal distribution

$$q(b_{k+1}|b^{(i)}_j, \ldots, b^{(i)}_k, \mathbf{y}) = p(b_{k+1}|b^{(i)}_k, \mathbf{y}) \qquad (4)$$

While it is intractable to compute Equation 4 exactly, it is straightforward to estimate using ACMI's Markov-field model

$$\begin{aligned} p(b_{k+1}|b^{(i)}_k, y_k) &\propto p(b^{(i)}_k, b_{k+1}|\mathbf{y})/p(b^{(i)}_k|\mathbf{y}) \\ &= \hat{p}_{k+1}(b_{k+1}) \cdot \psi_{adj}(b^{(i)}_k, b_{k+1}) \qquad (5) \end{aligned}$$

Here, $\hat{p}_{k+1}(b_{k+1})$ is the ACMI-computed marginal distribution for amino-acid $k+1$ ($\hat{p}_{k+1}$'s dependence on $\mathbf{y}$ dropped for clarity). We sample Cα $k+1$'s location from the product of (a) $k+1$'s marginal distribution and (b) the adjacency potential between Cα k and Cα $k+1$.

The optimal sampling function has a corresponding weight update

$$w^i_{k+1} \propto w^i_k \times \int p(y_{k+1}|b_{k+1}, b^{(i)}_k)\, db_k \qquad (6)$$

This integral, too, is intractable to compute exactly, but can be approximated using ACMI's marginals

$$w^i_{k+1} \propto w^i_k \times \int \hat{p}_{k+1}(b_{k+1}) \cdot \psi_{adj}(b^{(i)}_k, b_{k+1})\, db_k \qquad (7)$$

Equations 5 and 7 suggest a sampling approach to the problem of choosing location of Cα $k+1$ and reweighting each particle. This sampling approach, shown in Algorithm 1, is illustrated pictorially in Figure 4.

We sample M *potential* Cα locations from $\psi_{adj}(b^{(i)}_k, b_{k+1})$, the adjacency potential between k and $k+1$, which models the allowable conformations between two adjacent Cα's. We assign each sample a weight: the approximate marginal probability $\hat{p}_{k+1}$ at each of these sampled locations. We select a sample from this weighted distribution, approximating Equation 5. Finally, we reweight our particle as the sum of weights of all the samples we considered. This sum approximates the integral in Equation 7.

This process, in which we consider M potential Cα locations, is repeated for every particle in the particle filter for each Cα in the protein. For every particle, we begin by sampling locations for the amino-acid k whose marginal distribution has the lowest entropy (we use a soft-minimum to introduce randomness in the order in which amino acids are placed). This corresponds to the amino acid which ACMI is most sure of the location. The direction we sample at each iteration (i.e. toward the N- or C-terminus) is also decided by the entropy of the marginal distributions.

3.2.2 Using sidechain templates to sample sidechains Once our particle filter has placed Cα's $k-1$, k, and $k+1$ at 3D locations $b^{(i)}_{k-1:k+1}$, it is ready to place all the sidechain atoms in amino-acid k. We denote the position of these sidechain atoms s_k. Given the primary amino-acid sequence around k, we consider all previously observed conformations (i.e., those in the PDB) of sidechain k. Thus, s_k consists of (a) an index into a database of known sidechain 3D structures and (b) a rotation.

To further simplify, each sidechain template models the position of every atom from Cα^{k-1} to Cα^{k+1}. Then, given three consecutive backbone positions $b^{(i)}_{k-1:k+1}$, the orientation of sidechain s_k is determined by aligning the three Cα's in the sidechain template to $b^{(i)}_{k-1:k+1}$.

As Algorithm 1 shows, sidechain placement is quite similar to the Cα placement in the previous section. One key difference is that sidechain placement cannot take advantage of ACMI's marginal distribution, as ACMI's probability distributions have marginalized away sidechain conformations. Instead, the probability of a sidechain is calculated on-the-fly using the correlation coefficient between a potential conformation's calculated density and a region around b_k in the density map.

Figure 5 illustrates the process of choosing a sidechain conformation for a single particle i. We consider each of L different sidechain conformations for amino-acid k. For each sidechain conformation s^{*l}_k, $l \in \{1, \ldots, L\}$, we compute the correlation coefficient between the conformation and the map

$$CC^l = \text{cross-correlation}(s^{*l}_k, \text{EDM}[b^{(i)}_k])$$

$\text{EDM}[b_k]$ denotes a region of density in the neighborhood of b_k.

To assign a probability $p(\text{EDM}[b^{(i)}_k]|s_k)$ to each sidechain conformation, we compute the probability that a cross-correlation value was not generated by chance. That is, we assume that the distribution of the cross correlation of two random functions is normally distributed with mean μ and variance σ^2. We learn these parameters by computing correlation coefficients between randomly sampled locations in the map. Given some cross correlation x_c, we compute the expected probability that we would see score x_c or higher by random chance,

$$p_{null}(x_c) = P(X \geq x_c; \mu, \sigma^2) = 1 - \Phi(x_c - \mu/\sigma) \qquad (8)$$

Here, $\Phi(x)$ is the normal cumulative distribution function. The probability of a particular sidechain conformation is then

$$p(\text{EDM}[b^{(i)}_k]|s^{*l}_k) \propto (1/p_{null}) - 1 \qquad (9)$$

Since we are drawing sidechain conformations from the distribution of all solved structures, we assume a uniform prior distribution on sidechain conformations, so $p(s^{*l}_k\,|\,\text{EDM}[b^{(i)}_k]) \propto p(\text{EDM}[b^{(i)}_k]|s^{*l}_k)$.

As illustrated in Figure 5, sidechain sampling uses a similar method to the previous section's backbone sampling. We consider extending our particle by each of the L sidechain conformations $\{s^{*1}_k, \ldots, s^{*L}_k\}$ sampled from our

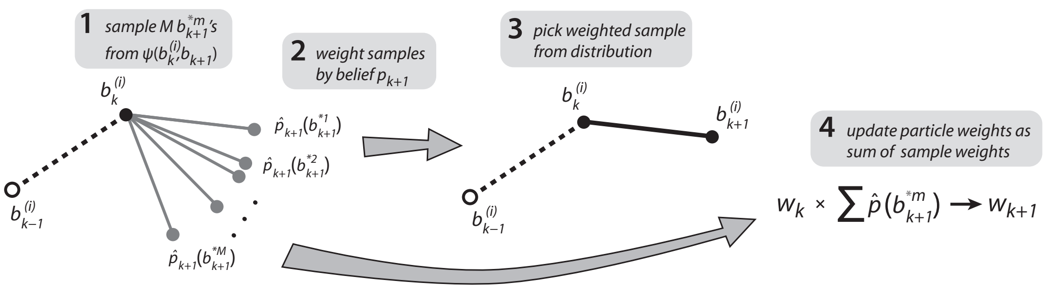
1 sample M b^{*m}_{k+1}'s from $\psi(b^{(i)}_k, b_{k+1})$
2 weight samples by belief p_{k+1}
3 pick weighted sample from distribution
4 update particle weights as sum of sample weights
$w_k \times \sum \hat{p}(b^{*m}_{k+1}) \rightarrow w_{k+1}$

Fig. 4. An overview of the backbone forward-sampling step. Given positions b_{k-1} and b_k, we sample M positions for b_{k-1} using the empirically-derived distribution of Cα–Cα–Cα pseudoangles. Each potential b_{k+1} is weighted by the belief $\hat{p}(b^{*m}_{k+1}|\mathbf{y})$. We choose a single location from this distribution; the particle weight is multiplied by the *sum* of these weights in order to approximate Equation 6.

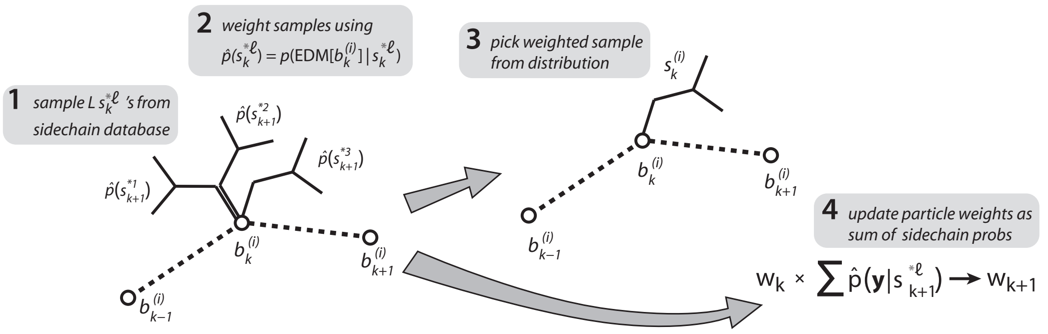
1 sample L $s^{*\ell}_k$'s from sidechain database
2 weight samples using $\hat{p}(s^{*\ell}_k) = p(\text{EDM}[b^{(i)}_k] \mid s^{*\ell}_k)$
3 pick weighted sample from distribution
4 update particle weights as sum of sidechain probs
$w_k \times \sum \hat{p}(\mathbf{y}|s^{*\ell}_{k+1}) \rightarrow w_{k+1}$

Fig. 5. An overview of the sidechain sampling step. Given positions $b_{k-1:k+1}$, we consider L sidechain conformations s^{*l}_k. Each potential conformation is weighted by the probability of the map given the sidechain conformation, as given in Equation 9. We choose a sidechain from this distribution; the particle weight is multiplied by the *sum* of these weights.

sidechain database. After computing the correlation between each sidechain conformation's density and the density map around b_k, each conformation is weighted using Equation 9. We choose a single conformation at random from this weighted distribution, updating each particle's weight by the *sum* of weights of all considered sidechain conformations.

Finally, our model takes into account the partial model $x_{j:k-1}$ when placing sidechain s_k. If any atom in sidechain s_k overlaps a previously placed atom or any symmetric copy, particle weight is set to zero.

3.3 Crystallographic data

Ten experimentally phased electron-density maps, provided by the Center for Eukaryotic Structural Genomics (CESG) at UW–Madison, have been used to test ACMI-PF. The maps were initially phased using AUTOSHARP (Terwilliger, 2002), with non-crystallographic symmetry averaging and solvent flattening (in RESOLVE) used to improve the map quality where possible. The ten maps were selected as the "most difficult" from a larger dataset of twenty maps provided by CESG. These structures have been previously solved and deposited to the PDB, enabling a direct comparison with the final refined model. All ten required a great deal of human effort to build and refine the final atomic model.

The data are summarized in Table 1, with quality described by the resolution and phase error. The resolution from the initial phasing may not have reached the resolution limit of the data set. Initial low-resolution phasing was computationally extended in three structures (using an algorithm in RESOLVE). Mean phase error was computed using CCP4 (Collaborative Computational Project, 1994) by comparing calculated phases from the deposited model with those in the initially phased dataset.

3.4 Computational Methodology

Models in ACMI-PF are built in three phases: (a) prior distributions are computed, (b) ACMI infers posterior distributions for each Cα location, and (c) all-atom models are constructed using particle filtering. Where available, ACMI used the location of selenium atom peaks as a soft constraint on the positions of methionine residues. Particle filtering was run ten times; in each run, the single highest-weight model was returned, producing a total of ten ACMI-PF protein models. Predicted models were refined for 10 iterations using REFMAC5 (Murshudov *et al.*, 1997), with no modification or added solvent. The first step is the most computationally expensive, but is efficiently divided across multiple processors. Computation time varied depending on protein size; the entire process took at most a week of CPU time on ten processors.

We compare ACMI-PF to four different approaches using the same ten density maps. To test the utility of the particle-filtering method for building all-atom models, we use the structure that results from independently placing

Table 1. Summary of crystallographic data.

PDB ID	AAs in ASU	Molecules in ASU	Resolution (Å)	Phase error (°)[a]
2NXF[b]	322	1	1.9	58°
2Q7A[b]	316	2	2.6	49°
XXXX[d]	566	2	2.65	54°
1XRI	430	2	3.3	39°
1ZTP	753	3	2.5	42°
1Y0Z	660	2	2.4 (3.7[c])	58°
2A3Q	340	2	2.3 (3.5[c])	66°
2IFU	1220	4	3.5	50°
2BDU	594	2	2.35	55°
2AB1	244	2	2.6 (4.0[c])	66°

[a] averaged over all resolution shells
[b] different dataset was used to solve the PDB structure
[c] phasing was extended from lower resolution
[d] PDB file not yet released

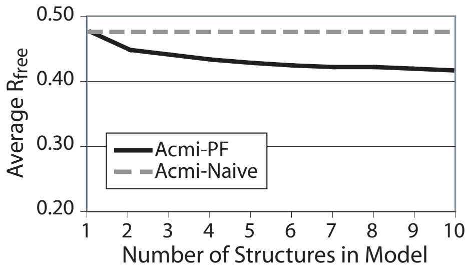

Fig. 6. A comparison of the R_{free} of ACMI-NAÏVE and ACMI-PF, as the number of protein models produced varies. Multiple models are produced by independent ACMI-PF runs (ACMI-NAÏVE only produces a single model). Since R_{free} in deposited structures is typically 0.20-0.25, we use 0.20 as the lowest value on the y-axis.

the best matching sidechain on each Cα predicted by ACMI, which we term ACMI-NAÏVE. The other three approaches are the commonly used density-map interpretation algorithms ARP/wARP (version 7), TEXTAL (in PHENIX version 1.1a), and RESOLVE (version 2.10). Refinement for all algorithms uses the same protocol as ACMI-PF, refining the predicted models for 10 iterations in REFMAC5 (ARP/wARP, which integrates refinement and model-building, was not further refined).

To assess the prediction quality of each algorithm, we consider three different performance metrics: (a) backbone completeness, (b) sidechain identification, and (c) R factor. The first metric compares the predicted model to the deposited model, counting the fraction of Cα's placed within 2 Å of *some* Cα in the PDB-deposited model. The second measure counts the fraction of Cα's both correctly placed within 2 Å *and* whose sidechain type matches the PDB-deposited structure. Finally, the R factor is a measure of deviation between the reflection intensities predicted by the model and those experimentally measured. A lower R factor indicates a better model. The R factor is computed using only peptide atoms (i.e., no added water molecules). The comparison uses the so-called free R factor (Brunger, 1992), which is based on reflections that were not used in refinement.

4 RESULTS AND DISCUSSION

4.1 ACMI-NAÏVE versus ACMI-PF

We first compare protein models produced by ACMI-PF to those produced by ACMI-NAÏVE. The key advantage of particle filtering is the ability to produce multiple protein structures using ensembles of particles. Since the density map is an average over many molecules of the protein in the crystal, it is natural to use multiple conformations to model this data. There is evidence that a single conformation is insufficient to model protein electron density (Burling & Brunger, 1994; Levin *et al.*, 2007; Furnham *et al.*, 2006; DePristo *et al*, 2004). As comparison, we take ACMI-NAÏVE, which uses the maximum-marginal trace to produce a single model.

We use ACMI-PF to generate multiple physically feasible models, by performing ten different ACMI-PF runs of 100 particles each. Each run sampled amino acids in a different order; amino acids whose belief had lowest entropy (i.e., those we are most confident we know) were stochastically preferred. Figure 6 summarizes the results. The y-axis shows the average (over the ten maps) R_{free} of the final refined model; the x-axis indicates the number of ACMI-PF runs. This plot shows that a single ACMI-PF model has an R_{free} approximately equal to the R_{free} of ACMI-NAÏVE. Model completeness is also very close between the two (data not shown). As additional structures are added ACMI-PF's model, average R_{free} decreases. The plot shows ACMI-NAÏVE's model

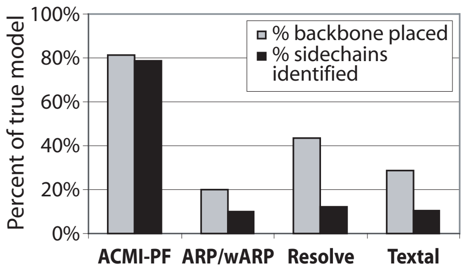

Fig. 7. A comparison of ACMI-PF to other automatic interpretation methods in terms of average backbone completeness and sidechain identification.

as a straight line, since there is no mechanism to generate multiple conformations. We believe a key reason for this result is that particle filtering occasionally makes mistakes when tracing the main chain, but it is unlikely for multiple PF runs to repeat the same mistake. The mistakes average out in the ensemble, producing a lower R factor.

Individual models in ACMI-PF offer additional advantages over ACMI-NAÏVE. Comparing the ACMI-PF model with lowest R_{work} (the "training set" R factor) to ACMI-NAÏVE's model shows that particle filter produces fewer chains on average (28 versus 10) and lower all-atom RMS error (1.60Å versus 1.72Å). This trend held in all ten maps in our test set: ACMI-PF's best model contains fewer predicted chains and lower RMS error than ACMI-NAÏVE. Additionally, the structures particle filtering returns are physically feasible, with no overlapping sidechains or invalid bond lengths.

4.2 Comparison to other algorithms

We further compare the models produced by particle filtering on the ten maps to those produced by three other methods for automatic density-map interpretation, including two well-established lower-resolution algorithms, TEXTAL and RESOLVE, and the atom-based ARP/wARP (although most of our maps are outside of its recommended resolution).

Figure 7 compares all four methods in terms of backbone completeness and sidechain identification, averaged over all ten structures. To provide a fair comparison, we compute completeness of a single ACMI-PF structure (of the ten produced). The ACMI-PF model chosen was that with the lowest refined R_{work}. Under both of these metrics, ACMI-PF locates a greater fraction of the protein than the other approaches. ACMI-PF performs particularly well at sidechain identification, correctly identifying close to 80% of sidechains over these ten poor-quality maps. The least accurate model that ACMI-PF generated (for 2AB1) had 62% backbone completeness and 55% sidechain identification. In contrast, the three comparison methods all return at least five structures with less than 40% backbone completeness and at least eight structures with less than 20% sidechain identification.

Scatterplots in Figure 8 compare the R_{free} of ACMI-PF's complete (10-structure) model to each of the three alternative approaches, for each density map. Any point below the diagonal corresponds to a map for which ACMI-PF's solution has a lower (i.e., better) R_{free}. These plots show that for all but one map ACMI-PF's solution has the lowest R factor. The singular exception for which ARP/wARP has a lower R factor is 2NXF, a high (1.9Å) resolution but poorly phased density map in which ARP/wARP automatically traces 90%, while ACMI-PF's best model correctly predicts only 74%. Our results illustrate both the limitations and the

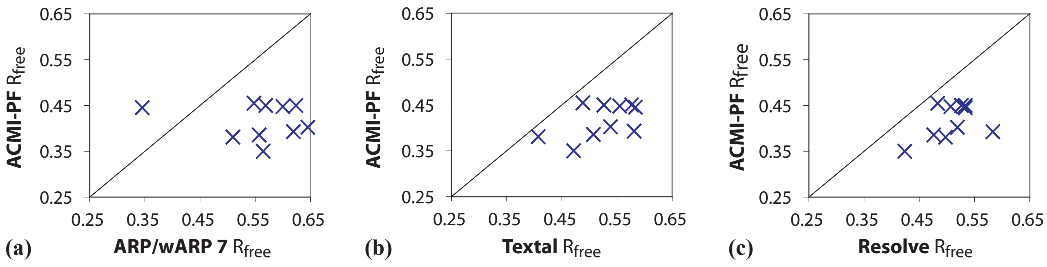

Fig. 8. A comparison of the free R factor of ACMI-PF's interpretation for each of the ten maps versus (a) ARP/wARP, (b) TEXTAL, and (c) RESOLVE. The scatterplots show each interpreted map as a point, where the x-axis measures the R_{free} of ACMI-PF and the y-axis the alternative approach.

advantages of ACMI-PF: it is consistently superior at interpretation of poorly phased, lower resolution maps, while an iterative phase-improvement algorithm like ARP/wARP may be better suited for a poorly phased but higher-resolution data.

5 CONCLUSION

We develop ACMI-PF, an algorithm that uses particle filtering to produce a set of all-atom protein models for a given density map. Particle filtering considers growing stepwise an ensemble of all-atom protein models. The method builds on our previous work, where we infer a probability distribution of each amino acid's Cα location. ACMI-PF addresses shortcomings of our previous work, producing a set of physically feasible protein structures that best explain the density map. Our results indicate that ACMI-PF generates more accurate and more complete models than other state-of-the-art automated interpretation methods for poor-resolution density map data. ACMI-PF produces accurate interpretations, on average finding and identifying 80% of the protein structure in poorly phased 2.5 to 3.5 Å resolution maps.

Using ACMI-PF, an ensemble of conformations may be easily generated using multiple runs of particle filtering. We show that sets of multiple structures generated from multiple particle filtering runs better fit the density map than a single structure. This is consistent with recent observations of the inadequacy of the single-model paradigm for modeling flexible protein molecules (Burling & Brunger, 1994; Furnham *et al.*, 2006; DePristo *et al*, 2004) and with the encouraging results of the ensemble refinement approach (Levin *et al.*, 2007). The ensemble description may also provide valuable information about protein conformational dynamics. As well, multiple conformations may be valuable for application of ACMI-PF in an iterative approach, where computed phases from an ACMI-PF model are used build an updated density map, which is fed back into the ACMI pipeline.

ACMI-PF's model-based approach is very flexible, and allows integration of multiple sources of "fuzzy" information, such as locations of selenium peaks. In the future, it may be productive to integrate other sources of information in our model. A more complicated reweighting function based on physical or statistical energy could better overcome ambiguities of unclear regions in the density map. The inclusion of these and other sources of information is possible, so long as they can be expressed in the probabilistic framework proposed here. This could further extend the resolution in which automated interpretation of density maps is possible.

ACKNOWLEDGEMENT

We acknowledge support from NLM T15-LM007359 (FD,AS,DK), NLM R01-LM008796 (FD,JS,GP,DK), and NIH Protein Structure Initiative Grant GM074901 (EB,CB,GP).

REFERENCES

Arulampalam,M.S., Maskell,S., Gordon,N., and Clapp,T. (2001). A tutorial on particle filters. *IEEE Trans. of Signal Processing*, 50, 174-188.

Berman,H.M. and Westbrook,J.D. (2004). The impact of structural genomics on the protein data bank. *Am. J. Pharmacogenomics*, 4, 247-252.

Brunger,A.T. (1992). Free R value: A novel statistical quantity for assessing the accuracy of crystal structures. *Nature*, 355, 472-475.

Burling,F.T. and Brunger,A.T. (1994). Thermal motion and conformational disorder in protein crystal-structures – comparison of multi-conformer and time-averaging models. *Israel J. of Chemistry*, 34, 165-175.

Chandonia,J.M. and Brenner,S.E. (2006). The impact of structural genomics: Expectations and outcomes. *Science*, 311, 347-351.

Collaborative Computational Project, Number 4 (1994). The CCP4 suite: Programs for protein crystallography. *Acta Cryst.*, D50, 760-763.

Cowtan,K. (2006). The Buccaneer software for automated model building.1.Tracing protein chains. *Acta Cryst.*, D62, 1002-1011.

DePristo,M.A., de Bakker,P.I., and Blundell, T.L. (2004). Heterogeneity and inaccuracy in protein structures solved by X-ray crystallography. *Structure*, 12, 911-917.

DiMaio,F., Shavlik,J.W., and Phillips,G.N.,Jr. (2006). A probabilistic approach to protein backbone tracing in electron-density maps. *Bioinformatics*, 22, e81-e89.

DiMaio,F., Soni,A., Phillips,G.N,Jr., and Shavlik,J.W. (2007). Improved methods for template-matching in electron-density maps using spherical harmonics. *Proc. IEEE Conf. on Bioinformatics and Biomedicine*, Fremont, CA.

Doucet,A., Godsill,S., and Andrieu,S. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statist. Comp*, 10, 197-208.

Furnham,N., Blundell,T.L., DePristo,M.A., and Terwilliger,T.C. (2006). Is one solution good enough? *Nature Struct. & Mol. Biol.*, 13, 184-185.

Geman,S. and Geman,D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. of PAMI*, 6, 721-741.

Ioerger,T.R. and Sacchettini,J.C. (2002). Automatic modeling of protein backbones in electron density maps. *Acta Cryst.*, D58, 2043-2054.

Ioerger,T.R. and Sacchettini,J.C. (2003). The TEXTAL system: Artificial intelligence techniques for automated protein model building. *Meth. Enz.*, 374, 244-270.

Kong,A., Liu,J.S., and Wong,W.H. (1994). Sequential imputations and Bayesian missing data problems. *J. Amer. Stat. Assoc*, 89, 278-288.

Levin,E.J., Kondrashov,D.A., Wesenberg,G., and Phillips,G.N.,Jr. (2007). Ensemble refinement of protein crystal structures. *Structure*, in press.

Morris,R., Perrakis,A. and Lamzin,V.S. (2003). ARP/wARP and automatic interpretation of protein electron density maps. *Meth. Enz.*, 374, 229-244.

Murshudov,G.N., Vagin,A.A., and Dodson,E.J. (1997). Refinement of macromolecular structures by the maximum-likelihood method. *Acta Cryst.*, D53, 240-255.

Sawasaki,T., Ogasawara,T., Morishita,R., and Endo,Y. (2002). A cell-free protein synthesis system for high-throughput proteomics. *PNAS*, 99, 14652-14657.

Snell,G. *et al.* (2004). Automated sample mounting and alignment system for biological crystallography at a synchrotron source. *Structure*, 12, 537-545.

Terwilliger,T.C. (2002). Automated main-chain model-building by template-matching and iterative fragment extension. *Acta Cryst*, D59, 38-44.

Wang,G. and Dunbrack,R.L. (2003). PISCES: A protein sequence culling server. *Bioinformatics*, 19, 1589-1591.